View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Springer - Publisher Connector

Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 56928, 13 pages
doi:10.1155/2007/56928

Research Article

Image and Video Indexing Using Networks of Operators

Stéphane Ayache,[1] Georges Quénot,[1] and Jérôme Gensel[2]

[1] *Multimedia Information Retrieval (MRIM) Group of LIG, Laboratoire d'Informatique de Grenoble, 385 rue de la Bibliothèque, B.P. 53, 38041 Grenoble, Cedex 9, France*
[2] *Spatio-Temporal Information, Adaptability, Multimédia and Knowledge Représentation (STEAMER) Group of LIG, Laboratoire d'Informatique de Grenoble, 385 rue de la Bibliothèque, B.P. 53, 38041 Grenoble, Cedex 9, France*

Received 28 November 2006; Revised 9 July 2007; Accepted 16 September 2007

Recommended by M. R. Naphade

This article presents a framework for the design of concept detection systems for image and video indexing. This framework integrates in a homogeneous way all the data and processing types. The semantic gap is crossed in a number of steps, each producing a small increase in the abstraction level of the handled data. All the data inside the semantic gap and on both sides included are seen as a homogeneous type called *numcept* and all the processing modules between the various numcepts are seen as a homogeneous type called *operator*. Concepts are extracted from the raw signal using networks of operators operating on numcepts. These networks can be represented as data-flow graphs and the introduced homogenizations allow fusing elements regardless of their nature. Low-level descriptors can be fused with intermediate of final concepts. This framework has been used to build a variety of indexing networks for images and videos and to evaluate many aspects of them. Using annotated corpora and protocols of the 2003 to 2006 TRECVID evaluation campaigns, the benefit brought by the use of individual features, the use of several modalities, the use of various fusion strategies, and the use of topologic and conceptual contexts was measured. The framework proved its efficiency for the design and evaluation of a series of network architectures while factorizing the training effort for common subnetworks.

Copyright © 2007 Stéphane Ayache et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Indexing image and video documents by concepts is a key issue for an efficient management of multimedia repositories. It is necessary and also a very challenging problem because, unlike in the case of the text media, there is no simple correspondence between the basic elements (the numerical values of image pixels and/or of audio samples) and the information (typically concepts) useful to users for searching or browsing. This is usually referred to as the *semantic gap* (between *signal* and *semantics*) problem.

The first thing that is commonly done for bridging the semantic gap is to extract *low-level descriptors* (that may be 3D color histograms or Gabor transforms, e.g.) and then extract concepts from them. However, even doing so, most of the semantic gap is still there (in the second step). The correlation between the input (low-level features) and the output (concepts) is still too weak to be efficiently recovered using a single "flat" classifier, even if the low-level features are carefully chosen.

The second thing that can be done is to split the concept classifier into two or more layers. *Intermediate entities* can be extracted from the low-level features (or from other intermediate entities) and the concepts can then be extracted from the intermediate entities (and possibly also from the low-level features). This approach is now widely used for concept detection in video documents [1–9] by the means of a stacking technique [10]. This approach performs better than the "flat" one probably because the correlations between the inputs and the outputs of each layer are much stronger than between the inputs and the outputs of the overall system. Then, even if the errors may accumulate across the layers, the overall performance may be increased if all layers perform much better than the flat solution. Furthermore, the system might not only be a linear series of classifiers (or other type of operators like fusion modules), it might also be a complex and irregular network of them.

In order to increase the performance of the indexing systems, more and more features and more and more layers are inserted. The considered networks become more and more complex and heterogeneous, especially if we include within them the feature extraction and/or the media decompression stages. The heterogeneity becomes greater considering both the handled data and the processing modules. Also, the

status of the intermediate entities as related either to signal or to semantics becomes less and less clear. This is why we propose a unified framework that hides the unnecessary heterogeneities and distinctions between them and keeps only one type of entity covering everything from media samples to concepts (included) and one type of processing module also covering everything from decompressors or feature extractors to classifiers or fusion modules. In the following, we call these entities and modules *numcepts* and *operators*. This approach also allows describing and manipulating the networks of heterogeneous operators using a *functional programming* (FP) style [11, 12].

For image and video indexing, many visual and text features have been considered. The text input may come from the context (if the image appears within a web page, e.g.) or from associated metadata. In the case of video, it may come from speech transcription using ASR or from closed captions when available.

On the visual side, *local* and *global* features can be used. Local features are associated to image parts (small patches or regions obtained by automatic image segmentations, e.g.) while global features are associated to the whole image. Local features usually appear several times within an image descriptions. Local and global visual features can represent various aspects of the image or video contents (color and texture, e.g.) and in different ways for local and global descriptions. The use of local features allows representing the *topological* context for the occurrence of a given concept like in discriminative random fields [13], for instance. Another source of context for the detection of a concept is the result of the detection of other concepts [14], which we call the *conceptual* context.

On the textual side, different features may also be considered like word distributions or occurrences of named entities. We introduced a new one which we call "topic concepts" [15] which is related to the detection of predefined categories.

The most successful approaches (cited above) tend to use features as varied as possible and as numerous as possible. They also tend to use the available contexts as mush as possible through the ways these features are combined. There are many ways to choose which features to combine with which other features and many ways to choose how to combine them. These combinations, usually called *fusion* can be done according to various strategies, the most common ones uses the *early* and *late* schemes [16]. We also introduced the *kernel* fusion scheme for concept indexing in video documents [17] which is applicable to the case of kernel-based classifiers like support vector machines (SVMs) [18].

The NIST TRECVID benchmark [19] created a task dedicated to the evaluation of the performance of concept detection. In the 2005 and 2006 editions, the concepts to be detected were selected within the large-scale concept ontology for multimedia (LSCOM) [20].

In this paper, we present the numcepts and operators framework and several experiments that we conducted within its context. In Sections 2 and 3, we present the framework and some application examples. In Section 4, we present experiments using the topological and conceptual contexts and, in Section 5, we present experiments using the "topic concepts." In both cases, the relative performances of the various features and of their combinations using various fusion strategies are compared in the context of the TRECVID benchmarks. Finally, in Section 6, we present the results obtained in the official TRECVID 2006 evaluation.

2. NUMCEPTS AND OPERATORS

Numcepts are introduced for clarifying, generalizing, and unifying several concepts used in information processing between the digital (or signal) level and the conceptual (or semantic) level. We find that there are many types of objects like signal, pixels, samples, descriptors, character strings, features, contours, regions, blobs, points of interests, shapes, shading, motion vectors, intermediate concepts, proto-concepts, patch-concepts, percepts, topics, concepts, relations, and so forth. All of them are not exclusive and their meaning may differ according to authors. This is amplified in the context of approaches using layers or networks (inspired from "stacking" [10] and currently the most efficient) that make use of *intermediate entities* that are no longer clearly either numerical descriptors in the classical sense or concepts also in the classical sense (i.e., something having a meaning for a human being).

The *numcept* term is derived from the *number* (or *num*erical description) and con*cept* (or *conceptual* description) terms and it aims at describing something that generalizes and unifies these two types of things that are often considered as qualitatively different. Indeed, one of the main difficulties in bridging the semantic gap comes from the difference of nature that one intuitively perceives between these two types of information or levels, traditionally called *signal level* and *semantic level*.

From the computer point of view (i.e., from the point of view of an information processing system), such a qualitative difference does not actually exist. All the considered elements, whatever their abstraction level, are represented in a digital form (using numbers). This is only the way in which a human being will interpret these elements that can produce a qualitative difference between them. Indeed, one will always recognize as numerical image pixels or audio samples and one will always recognize as conceptual some output given at the other extremity of the information processing chain like the labels of the various concepts seen in an image (or the association of binary or real values to these labels).

If the system goes directly from the beginning (e.g., image pixels) to the end (e.g., probability of appearance of visual concepts) in a single step through a "black box" type classifier (either from the raw signal or from preprocessed signal, Gabor transform or three-dimensional color histogram of it, e.g.), the case is quite clear: the semantic gap is crossed (with a certain probability) in a single step and the numerical or conceptual status of what comes in and goes out of it is also clear. There is no problem in seeing a difference of nature between them.

On the other hand, if the system goes from the beginning to the end in several steps with black boxes placed serially or arranged in a complex network, possibly even including feedbacks, the numerical or conceptual status of the various

elements that circulate on the various links between the black boxes becomes less clear. There are still clearly numerical and clearly conceptual descriptions at both ends, possibly also in the few first of the few last layers, but it may happen that what is present in the most intermediate levels does not clearly fall in one or the other category. That may be the case, for instance, if what is found at such intermediate level is the result of an automatic clustering process (that may produce or not or in a disputable way clusters that are meaningful to human beings). That may also be the case for what have been defined as "intermediate concepts," "percepts" or "protoconcepts" in some approaches. It is then no longer possible to clearly identify the black boxes across which the semantic gap has been crossed. The introduction of a formal intermediate level does not help much, the fuzziness of the frontiers between the levels remains.

Rather than considering and formalizing several qualitative differences like signal level, intermediate level, semantic level, or still others, we propose instead to ignore any such qualitative difference and to consider them as irrelevant for our problems. *Numcepts* are the only type of objects that will be manipulated from the beginning to the end (and including the beginning and the end). Similarly and to keep coherence, we propose to consider only *operators* or *modules* taking as inputs only numcepts and producing as outputs only numcepts and to ignore any possible qualitative difference among them. Decompressors, descriptor extractors, supervised or unsupervised classifiers, fusion modules, and so forth will all appear as operators, whatever their level of abstraction and however they are actually implemented.

While doing these types of unification, we have made little progress from the practical point of view but we nevertheless moved from a heterogeneous approach to an homogeneous approach and we got rid of the rigidities of approaches layered according to predefined schemes (e.g., classifying the processing in low, middle, and high levels). This way of seeing things does not radically change things but it offers more flexibility and freedom in the design and the implementation of concept indexing systems. It permits to consider rich and varied architectures without thinking about the type of data handled or about the type of operator used. Any combination of data and operator type becomes possible and subject to experimental exploration. A numcept may be defined only by the way it is produced (computed or learned) from other numcepts and its use may be justified only by the gain in performance it is able to bring when introduced in an indexing system and this without having to wonder about its possible semantic level or about what it may actually represent or mean. A (partially) blind approach similar to natural selection becomes possible at all the levels of the system, equally for numcepts, for operators, and for the network architecture.

The considered systems are still designed for semantic indexing: as a whole they still take as inputs the numerical values of image pixels and/or audio samples, for instance, and they produce also numerical values that are associated to labels that (generally) correspond to something having a meaning for a human being. Also, this does not require that we forget everything we know about what has already been tried and identified as useful in the context of more rigid or heterogeneous approaches. These may be used as starting points, for instance. We may still consider the classical categories for various types of numcepts and operators whenever this appears possible and useful but we will ignore them and we will not be limited by them when they make little sense or imply unnecessary restrictions.

From a practical point of view, numcepts always are numerical structures. They can be either scalars or vectors or multidimensional arrays. They can also be irregular structures like sets of points of interest. The details of the practical implementation are not much relevant to the approach. The important point is that numcepts can have some types and that the operators that use them as inputs or outputs have to be of compatible types (possibly supporting overloading). The most common type is the vector of real numbers. It may include scalars, vectors, and multidimensional arrays if these can be linearized without loss of useful information.

Operators may also be of many types regarding the way they process numcepts. They may be fully explicitly described like a Gabor transform for feature extraction or like a majority decision module for fusion. They also may be implicitly defined, typically by learning from a set of samples and a learning algorithm. This learning may be supervised (classifiers) or unsupervised (clustering tools). Finally, the description of operators may also include some parameters like the number of bins in color histograms, the number of classes in a clustering tool, or some thresholds.

The "numcepts and operators" approach becomes interesting when large and complex networks are considered. It is able to handle multimodality, multiple features, multiple scales (local, intermediate, and global for the visual modality), and multiple contexts. It is likely that a high level of complexity for the operator networks will be necessary to achieve a good accuracy for concept detection in open application areas. The increase in complexity will be a challenge because of the combinatorial explosion of the possibilities of choosing and combining numcepts and operators. In the context of this approach, the operator networks of themselves can be learned through automatic generation and evaluation using for instance genetic algorithms. There will be a need for powerful tools for describing, handling, executing and evaluating all these possible architectures. One possibility for that is to use the formalism of *functional programming* over numcepts and operators.

We did not implement yet the automatic generation and evaluation of operator networks but we did generate variations in a systematic way and evaluated them. Some of these experimentations are reported in the next two sections. More information can be found in [7, 15, 17, 21].

The "numcepts and operators" approach has similarities with other works that also makes use of low level and intermediate features to detect the high-level semantic concepts using classifiers and fusion techniques like, for instance, [5, 22]. Most of these works can be expressed within the "numcepts and operators" framework which is a generalization of them. The semantic value chain analysis [22], for instance, corresponds to a sequence of operators that focuses sequentially on the contents, the style, and the context

aspects in order to refine the classification. There are also some similarities in the details of the instantiation between this work and the networks that we experimented, especially for the content and context aspects. What the framework brings is a greater level of generality, a greater flexibility, and an environment for the generation, evaluation, and the selection of network architectures.

There are some similarities between the way such network operates and the way the human brain might operate: both are (or seem to be) constituted of modules arranged in networks, both begin by processing feature separately by modalities and separately within modalities (color, texture, and motion, e.g.), both fuse the results of feature extraction using cascaded layers and both somehow manipulate very different type of data with very different type of processing modules somehow using a quite uniform type of "packaging" for them. Moreover, the features that are selected in practice for the low-level layers are also quite similar both for the audio and image processing.

Figure 1 gives an example of a complex network that could be used for the detection of a complex concept. Such networks may be adapted for the concepts they target or they may be generic.

3. NUMCEPTS FOR IMAGE AND VIDEO INDEXING

We consider a variety of numcepts for the building of indexing networks. We chose them at several levels (low and intermediate) and for several modalities (image and text). Intermediate numcepts are built from low-level ones and using an annotated corpus (e.g., TRECVID/LSCOM or Reuters). The operators that generate these intermediate numcepts are based on support vector machines (SVMs) [18]. Low-level numcepts are themselves generated from the raw image or from the text signal by explicit operators (moments, histograms, Gabor transforms, or optical flow), some of them being parameterizable. Text itself comes from an automatic speech recognition (ASR) operator applied to the raw audio signal.

All the classifiers used in our experiments are SVM classifiers. We use the libsvm implementation [23]. We use RBF kernels, and their parameters are always automatically adjusted by a five-fold cross-validation on the training set.

3.1. Visual numcepts

Many visual features can be considered. We made some choices that may be arbitrary but they follow the main trends in the domain as they include both local and global image representations and the classical color, texture, and motion aspects. These choices have been made for a baseline system. The main goal here is to explore the use of context for concept indexing. We want to study and evaluate various ways of doing it by combining operators into networks. In further work, we plan to enrich and optimize the set and characteristics of low-level features, especially for video content indexing. Currently, we expect to obtain representative results from the current set of low-level features.

3.1.1. Local visual feature numcepts

Local visual feature numcepts are computed on image patches. The patch size has been chosen to be small enough to generally include only one visual concept and large enough so that there are not too many of them and so that some significant statistics can be computed within them. For MPEG-1 video images of typical size of 352×264 pixels, we consider 260 (20×13) overlapping patches of 32×32 pixels. For each image patch, the corresponding local visual feature numcept includes (low-level features) the following:

(i) 2 spatial coordinates (of the center of the patch in the image),

(ii) 9 color components (RGB means, variances, and covariances),

(iii) 24 texture components (8 orientations $\times$ 3 scales Gabor transform),

(iv) 7 motion components (the central velocity components plus the mean, variance, and covariance of the velocity components within the patch; a velocity vector is computed for every image pixel using an optical flow tool [24] on the whole image).

3.1.2. Global visual feature numcepts

Global visual feature numcepts are computed on the whole image. They include (low-level features) the following:

(i) 64 color components ($4 \times 4 \times 4$ color histogram),

(ii) 40 texture components (8 orientations $\times$ 5 scales Gabor transform),

(iii) 5 motion components (the mean, variance, and covariance of the velocity components within the image).

3.1.3. Local intermediate numcepts

Local intermediate numcepts are computed on image patches from the local visual feature numcepts. They are learned from images in which classical concepts have been manually annotated. Each of them learned using a single SVM classifier; for a given local intermediate numcept, the same classifier is applied to all the patches within the image. We selected 15 classical concepts that were learned using the manual local concept annotation from the TRECVID 2003 and 2005 collaborative annotation [25] that was cleaned up and enriched. These 15 concepts are Animal, Building, Car, Cartoon, Crowd, Fire, Flag-US, Greenery, Maps, Road, Sea, Skin_face, Sky, Sports, and Studio_background.

The local intermediate numcepts can be interpreted as local instances of the original classical concepts they have been learned from. They indeed can be used as a basis for the detection of the same concepts at the image level. However, they have been designed for a use in a broader context: they are intended to be used as a basis for the detection on many other concepts, related or not to the learned one, whether or not they are relevant to the targeted concepts and whether or not they are accurately recognized.

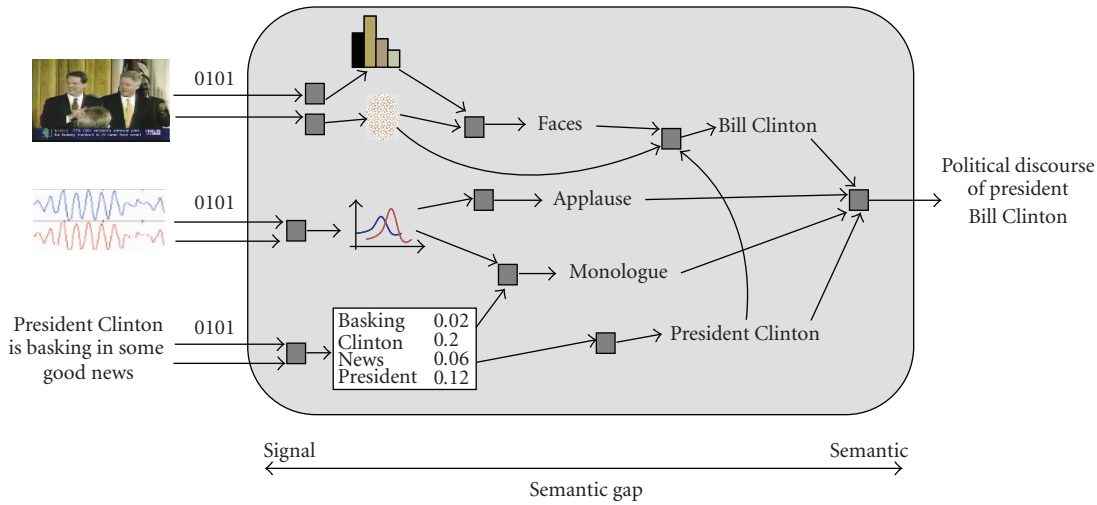

Figure 1: Example of a network of operators for the detection of a complex concept.

Local intermediate numcepts can be seen as a new raw material comparable to low-level features and that can be used as such for higher-level numcept extraction. From this respect, they have the advantage of being placed at a higher level inside the semantic gap as they are derived from something that had some meaning at the semantic level, even if what they are used for is not related to what they have been learnt from. They may somehow implicitly grasp some color/texture/motion/location combinations that are relevant beyond the original concepts which they are derived from. Another advantage is that a large number of concepts can be derived from a small number of them. This is quite efficient in practice since only the local intermediate numcepts need to be manually annotated at the region level (which is costly) while the targeted concepts only need to be annotated at the image level for learning.

When considered only as new raw material for higher level classification, local intermediate numcepts do not need to be accurately recognized. What is used in the subsequent layers is not the actual presence of the original concept but some learnt combination of the lower-level features. Poor recognition does not hurt the subsequent layers because they are trained using what has been learnt, not with what was supposed to be recognized. From their point of view, what is important is that the local intermediate numcepts are consistent between the training and test sets and that they grasp something meaningful in some sense.

3.2. Textual numcepts

Textual numcepts are derived from the textual transcription of the audio track of video documents which is obtained by automatic speech transcription (ASR) possibly followed by machine translation (MT) if the source language is not English. Text may also be extracted from the context of occurrence or from metadata. The textual numcepts are computed on audio speech segments as they come from the ASR output. Then, each video key frame is assigned the textual numcepts of the speech segment they fall into or those of the closest speech segment if do not all within one. Two types of text numcepts are considered. The first one is a low-level one is derived only from the raw text data. The second one is derived from the raw text data and from an external text corpus annotated by categories.

3.2.1. Text numcepts

Text numcepts are computed on audio segments of the ASR or ASR/MT transcription. A list of 2500 terms associated to a target concept is built considering the most frequent ones excluding stop words. A list is built for each final target concept. The text numcept is a vector of boolean values whose components are 0 or 1 if the term is absent or present in the audio segment.

3.2.2. Topic numcepts

Topic numcepts are derived from the speech transcription. We used 103 categories of the TREC Reuters (RCV1) collection [26] to classify each speech segment. The advantages of extracting such concepts from the Reuters collection are that they cover a large panel of news topics and they are obviously human understandable. Thus, they can be used for video search tasks. Examples of such topics are Economics, Disasters, Sports, and Weather. The Reuters collection contains about 810 000 text news items in the years 1996 and 1997.

We constructed a vector representation for each speech segment by applying stop-list and stemming. Also, in order to avoid noisy classification, we reduced the number of input terms. While the whole collection contains more than 250 000 terms, we have experimentally found that considering the top 2500 frequently occurring terms gives the better classification results on the Reuters collection. We built a prototype vector of each topic category on the Reuters collection and apply a Rocchio classification on each speech segment.

Such granularity is expected to provide robustness in terms of covered concepts as each speaker turn should be related to a single topic.

Our assumption is that the statistical distributions of the Reuters corpus and of target documents are similar enough to obtain relevant results. Like in the case of visual intermediate concepts, it is not necessary that these numcepts are accurately recognized or actually relevant for the targeted final concept. They can also be considered as new raw material and what is important is that the topic numcepts are consistent between the training and test sets and that they grasp something meaningful in some sense.

For each audio segment, the numcept is a vector of real values with one component per Reuters category. This value is the score of the audio segment for the corresponding category.

4. USE OF THE CONTEXT IN NETWORK OF OPERATORS

We conducted several experiments with various networks of classifiers. All the classifiers, including those used for fusion, were implemented with support vector machines (SVMs) [18] using the libsvm package [23]. We first tried networks that make use of topologic and semantic contexts. They are described here considering only the use of local visual features and/or with local intermediate numcepts.

Figure 2 shows the overall architecture of our framework and how classifiers are combined for the use of the topologic context and of the semantic context. Six different networks are actually shown in this figure and some of them share some subparts. The six outputs are numbered from 1 to 6. The first three make use only of the topologic context (Section 4.1), the last three make use of topologic and semantic contexts (Section 4.2).

4.1. Use of the topologic context

The idea behind the use of topologic context is that the confidence (or score) for a single patch (and for the whole image) could be computed more accurately by taking into account the confidences obtained for other patches in the image for the same concept. This idea has been used, for instance, in the work of Kumar and Hebert [13] and it could be used in a similar way within our framework. In our work, however, it is currently implemented only at the image level and this means that the decision at the image level is taken considering the set of the local decisions along with their locations.

We studied three network organizations to evaluate the effect of using the topologic context in concept detection at the image level. The first one is a baseline in which no context (either topologic or semantic) is used. The second one uses the topologic context in a flat (single layer) way while the third uses the topologic context in a serial (two layers) way.

In this part, we consider concepts independently one from another. Concept classifiers are trained independently from each other whatever their levels. In the following, N will be the number of concepts considered, P will be the number of patches used (260 in our experiments), and F will be the number of low-level feature vectors components (35 in our experiments, motion was not used there).

4.1.1. Baseline, no context, one level (1)

In order to evaluate the patch level alone, we define an image score based on the patch confidence values. To do so, we simply compute the average of all of the patch confidence scores. This baseline is very basic, it does not take into account any spatial or semantic context. We have here N classifiers, each with F inputs and 1 output. Each of them is called P times on a given image and the P output values are averaged.

4.1.2. Topologic context, flat, one level (2)

The "flat" network directly computes scores at the image level from feature vectors. We have here N classifiers, each with $F \times P$ inputs and 1 output. Each of them is called only once on a given image and the single output value is taken as the image score. This network organization is not very scalable and requires a lot of training data and training times because of the large number of inputs of the classifiers.

4.1.3. Topologic context, serial, two levels (3)

The "serial" network is similar to the baseline one. The difference is that the scores at the image level are computed by a second level of classifiers instead of averaging. We have here N level_1 classifiers, each with F inputs and 1 output and N level_2 classifiers, each with P inputs and 1 output. Each level_1 classifier is called P times on a given image and its P output values are passed to the corresponding level_2 classifier which is called only once. Topologic context is taken into account by concatenating patches confidence value in a vector.

4.2. Use of topologic and semantic contexts

We studied three other network organizations to evaluate the effect of using additionally the semantic context in concept detection at the image level. We still include outputs from the patch level, but we do so using the outputs related to all other concepts for the detection of any given concept. We are now considering concepts as related one to each other (and no longer independently one from another). The concept scores are combined using an additional level of SVM classifier (late fusion scheme).

4.2.1. Topologic and semantic contexts, sequential, three levels (4)

The fourth network simply takes the output of the third one (topologic context, serial, two levels) and adds a third level that uses the scores computed for all concepts to reevaluate the score of a given concept. We have additionally here N level_3 classifiers, each with N inputs and 1 output. Each level_3 classifier is called only once on a given image.

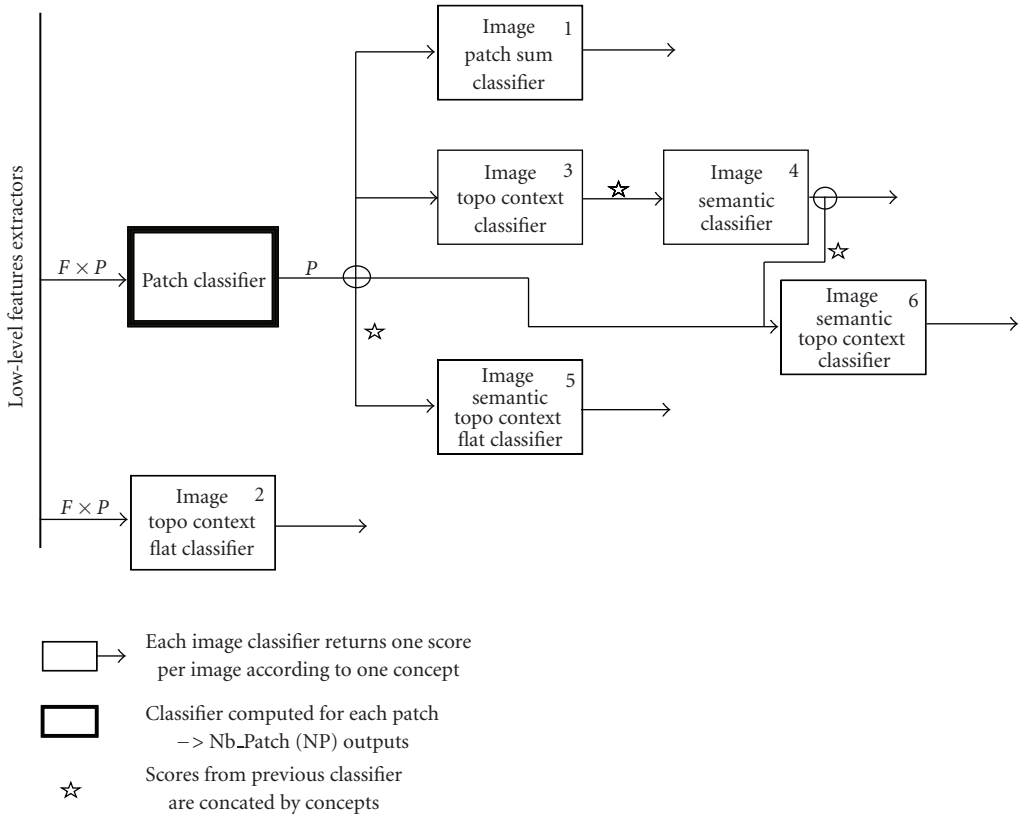

Figure 2: Networks of operators for evaluating the use of context.

4.2.2. *Topologic and semantic contexts, parallel, two levels (5)*

The fifth network is similar to the previous version except that the last two levels have been flattened and merged into a single classifier. The difference is similar to the difference between the serial and flat versions of the networks that use only the topologic context. We have here N level_1 classifiers, each with F inputs and 1 output and N level_2 classifiers, each with $N \times P$ inputs and 1 output. All level_1 classifiers are called P times on a given image and their $N \times P$ output values are passed to the corresponding level_2 classifier which is called only once.

4.2.3. *Topologic and semantic contexts, parallel, three levels (6)*

The previous network suffers from the same limitation as the other flattened version is not very scalable and requires a lot of training data and training times because of the large number of inputs of the classifiers. The flattening, however, permits to use the topologic and semantic information in parallel and in a correlated way. The sequential organization, on the contrary, though making use of both pieces of information does it in a noncorrelated way.

The sixth network organization tries to keep both contexts correlated (though less coupled) while avoiding the curse of dimensionality problem. The $N \times P$ number of inputs is replaced by $N + P$. The architecture is a kind of hybrid between the two previous ones. It is the same as in the sequential case but P inputs are added to the classifiers f the last level. These P inputs come directly from the output of the first level but for the corresponding concept only (instead of the output from all P patches times all N like in the flattened case).

5. FUSION USING NUMCEPTS AND OPERATORS

5.1. *Early and late fusion*

We consider here the early and late well-known fusion strategies as follows:

(i) *One-level fusion.* In a one-level fusion process, intermediate features or concepts are concatenated into a single flat classifier, as in an early fusion scheme [16]. Such a scheme takes advantage of the use of the semantic-topologic context from visual local concepts, and semantic context from topic concepts and visual global features. However, it is constrained by the curse of dimensionality problem. Also, the small numbers of topic concepts and global features compared to the huge amount of local concepts can be problematic: the final score might strongly depend upon the local concepts.

(ii) *Two-level fusion.* In a two-level fusion scheme, we classify high-level concepts from each modality separately

at a first level of fusion. Then, we merge the obtained outputs into a second-layer classifier. We investigate the following possible combinations. Classifying each high-level concept with intermediate classifiers then merging outputs into a second-level classifier is equivalent to the late fusion defined in [16]. Using more than two kinds of intermediate classifiers, we can also combine pair wise intermediate classifiers separately and then combine given scores in a higher classifier. For instance, we can first merge and classify global features with topic concepts and then combine the given score with outputs of local concept classifiers in a higher classifier. Another possibility is to merge separately local concepts with global features and local concepts with topic concepts, then to combine the given scores in a higher level classifier. Advantages of such schemes are numerous: the second-layer fusion classifier avoids the problem of unbalanced inputs, and keeps both topologic and semantic contexts at several abstraction levels.

These two fusion strategies can be used in several ways including a mix of both since we consider more than two types of input numcepts. We actually consider here four of them: "text," "topics," "local intermediate," and "global" numcepts as described in Section 3 (direct "local features" are not considered here). These numcepts are of different modalities (text and image) and of different semantic level (low and intermediate). We use the "$A - B$" notation for the early fusion of numcepts A and B and the "$A + B$" notation for the late fusion of numcepts A and B. We also use "lo," "gl," and "to" as short names for "local," "global," and "topic" numcepts, respectively.

Figure 3 shows the overall architecture of our framework and how classifiers are combined for evaluation of the various fusion strategies. Ten different networks are actually shown in this figure and several of them share some subparts. The ten outputs are labeled according to the way the fusion is done, as follows.

(i) First, the target concepts can be computed using only one type of numcept as input. In these cases, there is no fusion at all and the labels are simply the name of the used numcepts: "text," "topics," "local," and "global." These cases are defined to constitute baselines against which the various fusion strategies will be evaluated.

(ii) Second, early fusion schemes are used. Not all combinations are tried. The combinations are labeled using the first two letters of the fused numcepts separated by a minus sign that represents the early fusion of the classifiers that use them. The "lo-to," "gl–to," and "lo-gl–to" combinations have been selected.

(iii) Third, late fusion schemes are used, not only between the original numcepts but also between them and/or the numcepts resulting from an early fusion of them. Again, not all combinations are tried. The combination are labeled using the first two letters of the fused numcepts or the label of the previous early fusion separated by a plus sign that represent the late fusion of the classifiers that use them. The "lo+gl+to," "lo+gl–to," and "lo-to+gl–to" combinations have been selected (in this notation, the minus sign has precedence over the plus sign).

In Figure 3, F and P are, respectively, the number of low-level features computed on each image patch and the number of patches, G is the number of low-level features computed on the whole image, V is the number of local intermediate numcepts computed, N is the number of raw text features, and T is the number of topic numcepts computed.

5.2. Kernel fusion

In this part, we consider a third fusion scheme which is called "kernel fusion." It is intermediate between early and late fusion and offers advantages from both. It is applicable when classifiers are of the same type and based on the use of a kernel that combines sample vectors like SVM. A fused kernel is built by applying a combining function (typically sum or product) to the kernels associated to the different sources. The rest of the classifier remains the same [27].

6. EXPERIMENTS

The objective of this part of the work is to validate our assumptions and to quantify the benefits that can be obtained from various types of numcepts and from contextual information.

6.1. Evaluation of the use of the context

We conducted several experiments using the corpus developed in the TRECVID 2003 Collaborative Annotation effort [25] in order to study different fusion strategies over local visual numcepts. We used the trec_eval tool and TRECVID protocol, that is, return a ranked list of 2000 top images. The considered corpus contains 48 104 key frames. We split it into 50% training set and 50% test set.

We focus here on 5 concepts which can be extracted as patch-level: Building, Sky, Greenery, Skin_face, and Studio_Setting_Background. We choose them because of their semantics relationships. Building, Sky, Greenery are closer than others. Additionally, Skin_face and Studio_Setting_Background occur often together. In this part, the final targeted concepts are the same as those that have been used for the definition of the local intermediate numcepts.

We used SVM classifier with RBF Kernel, because it has shown good classification results in many fields, especially in CBIR [28]. We use cross-validation for parameter selection, using grid search tool to select the best combination of parameters C and gamma (out of 110).

In order to obtain the training set, we extracted patches from annotated regions; it is easy to get many patches by performing overlapped patches. Annotating whole images is harder as annotators must observe each one.

We collected many positive samples for patches annotation, and defined experimentally a threshold for maximum numbers of positive samples. We found that 2048 positive

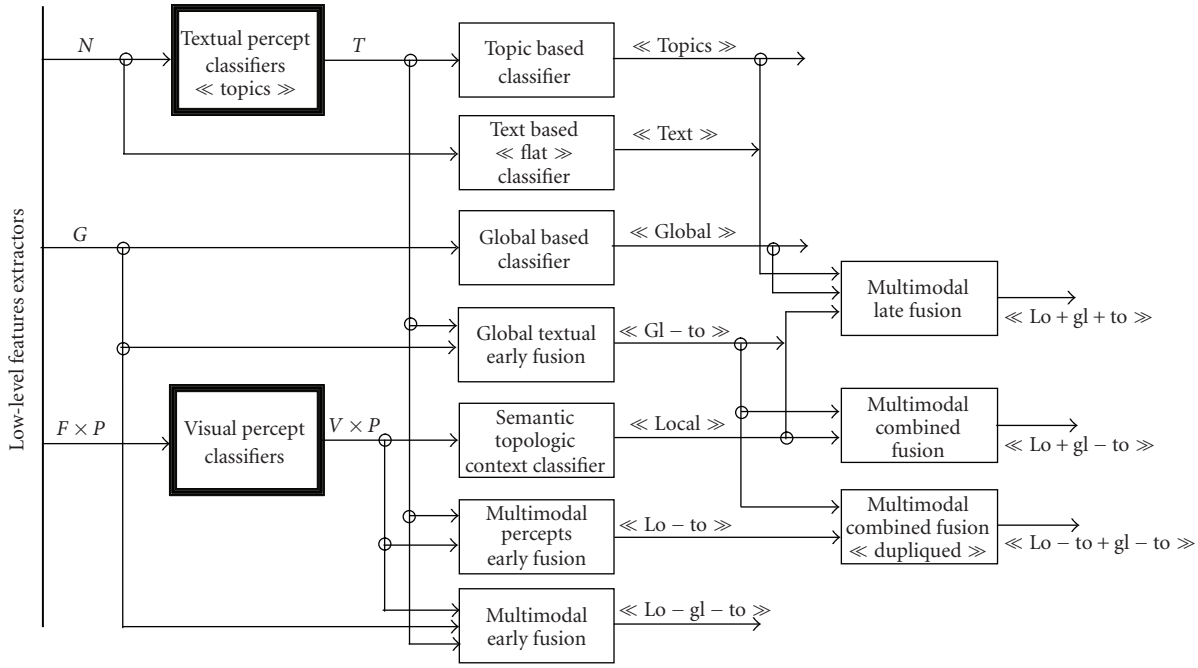

Figure 3: Networks of operators for evaluating fusion strategies.

samples is a good compromise to obtain good accuracy with smaller training time. Also, we found that using twice as many negative samples as positive samples is a good compromise. Finally, we randomly choose negative samples. Table 1 shows the number of positive image examples for each concept.

Table 1 shows the relative performance and training time for the detection of five concepts and for the six network organizations considered in Sections 4.1 and 4.2. As expected, the flattened version requires much more training time. For the presented times, we added the training times of each intermediate levels and included the cross-validation time. Also, the cross-validation process can be performed in parallel [23], we used 11 3 Ghz Pentium-4 processors. The reported results are for one single processor.

The use of topologic context improves the performance over the baseline and combined with the semantic context improves it even further. The performance of the three-level sequential classifier is poorer than the two-level serial one. This may be due to the lack of information of his final level classifier, which have N (currently 5) inputs only. This may change when a much higher number of concepts are used.

For the networks which use both topologic and semantic contexts, the hybrid version has an intermediate performance between the sequential and parallel flattened versions. The two-level version has the better performance as it merges more information. However, it does not scale well with the number of concepts while the hybrid version suffers much less from this limitation and should perform better with more concepts. Also, by comparing second and fifth networks results, we can conclude that dimensionality reduction induced by our approach is really significant, in term of both accuracy and computational time.

6.2. Evaluation of early and late fusion strategies

We have evaluated the use of visual and topic concepts and their combination for concepts detection in the conditions of the TRECVID 2005 evaluation. We show the 10 high-level concepts classification results evaluated with the trec_eval tool using the provided ground truth, and compare our results with the median over all participants. We have used a subset of the training set in order to exploit the speech transcription of the samples. As the quality of TRECVID 2005 transcription is quite noisy due to both transcription and translation from Chinese and Arabic videos, some video shots do not have any corresponding speech transcription. In order to compare visual only runs with topic concept based runs, we have trained all classifiers using only key frames whose transcript is not empty. In average, we have used about 300 positives samples and twice as many negative samples.

It has been shown in [26] that SVM outperforms a Rocchio classifier on text classification. In this experiment, we first show the improvement brought by the topic concepts based classification by comparing with an SVM text classifier based on the uttered speech occurring in a shot after same text analysis as topic classifiers. Then, we give some evidence of the relevance of using topic concepts, by showing the improvement of unimodal runs when combined with the topic concepts. In a second step, we compare one-level fusion with two-level fusion for combining intermediate concepts. We have implemented several two-level fusion schemes to merge the output of intermediate classifiers (Section 5.1). Particularly, we show that pair wise combinations schemes can increase high-level concepts classification.

We used an SVM classifier with RBF kernels as it has proved good performance in many fields, especially in

Table 1: Comparative performance of network organizations: mean average precision (MAP) for five concepts, mean of MAPS, and corresponding training times (in minutes).

	Build.	Sky	Stud.	Green.	Skin.	All	Time
Baseline, no context, one level (1)	0.341	0.161	0.409	0.626	0.158	0.339	396
Topologic context, flat, one level (2)	0.193	0.545	0.890	0.462	0.342	0.487	836
Topologic context, serial, two levels (3)	0.308	0.433	0.767	0.721	0.456	0.537	418
Topo. and semantic, sequential, three levels (4)	0.282	0.404	0.650	0.723	0.439	0.500	459
Topo. and semantic, parallel, two levels (5)	0.423	0.561	0.911	0.728	0.428	0.610	484
Topo. and semantic, parallel, three levels (6)	0.338	0.464	0.844	0.681	0.442	0.554	451
Number of positives, images, examples	383	1583	429	712	895	—	—

multimedia classification. LibSVM [23] implementation is easy to use and provides probabilistic classification scores as well as efficient cross validation tool. We have selected the best combination of parameters C and gamma out of 110, using the provided grid search tool.

Figure 4 shows the mean average precision (MAP) results of the conducted experiments. We compare our results with the TRECVID 2005 median result. The label of the runs corresponds to those of the networks described in Section 5.1.

Topic concepts based classification performs much better than text based classifier, the gain obtained by topic concepts based classification is obvious. It means that despite the poor quality of speech transcription, intermediate topic concepts are useful to reduce the semantic gap between uttered speech and high-level concepts. Each intermediate topic classifier provides significant semantic information despite the differences between Reuters and TRECVID transcripts corpora. It is interesting to notice that the Sports concept is also a Reuters category and has the best MAP value for the topic numcepts based classification.

For the "global" run, we have directly classified high-level concepts using their corresponding global low-level features. When combined with topic concepts, the average MAP increases by 30%, and up to 100% on Sports high-level concept. Also, some high-level concepts which have poor topic based classification MAP cannot benefit from the combination with topic concepts.

The use of the topologic-semantic context in local concepts based classification improves clearly the performance over the global based classifier. However, we observe a non significant gain when combined with topic concepts. This can be explained by the huge numbers of "local" inputs compared with the few numbers of "topic" inputs. Since we have used RBF kernel, the topic concepts inputs have a very small impact on the Euclidian distance between two examples. A solution to avoid such unbalanced inputs could be to reduce the numbers of local concepts inputs using a feature selection algorithm before merging with the topic concepts. Despite this observation, we notice that we obtain better results by combining "local" with "topic" concepts than combining "local" concepts with "global" features.

We have conducted several experiments to combine "topic" concepts with "local" and "global" features. Where "local" only classification performs very well for some "visual" high-level concepts (Mountain, Waterscape), we can observe an improvement using fusion based runs for most of high-level concepts.

The runs "lo-go-to" and "lo + go + to," which correspond, respectively, to the early and late fusion schemes, provide roughly similar results and do not outperform visual local classifier. This is probably due to the relative good performance of "local" run compared to other runs.

We have obtained the best results using a two-level fusion scheme combining separately topic concepts with local and global features in the first fusion layer. The "lo-to + go-to" mixed fusion scheme is an early fusion of the "topic" concepts with both "local" and "global" features separately followed by a late fusion. In this case, the duplication of topic concepts at the first level of fusion performs better by 10% than other fusion schemes. With such a scheme, topic concepts integrate useful context to visual features and achieve significant improvement, compared to unimodal classifiers, for most of high-level concepts.

6.3. Results TRECVID 2006

We participated to the TRECVID 2006 evaluation using several networks of operators. For each of the 39 concepts, we manually associated a subset of 5-6 intermediate visual numcepts. Thus, visual feature vectors contain about 1500 dimensions (5-6 $\times$ 260 local intermediate + 109 global low level).

Six official runs were submitted since this was the maximum allowed by TRECVID organizers but we actually prepared thirteen of them. The unofficial runs were prepared exactly in the same conditions and before the submission deadline. They are evaluated in the same conditions also using the tools and qrels (relevance judgments) given by the TRECVID organizers. The only difference is that they did not participate to the pooling process (which is statistically a slight disadvantage).

Table 2 gives the inferred average precision (IAP) of all our runs. The official runs are the numbered ones and the number corresponds to the run priority. The IAP of our first run is 0.088 which is slightly above the median while the best system had an IAP of 0.192.

The naming of the networks (and runs) is different here. The type of fusion (early, late, or kernel) is explicitly indicated in the name (no mixture of fusion schemes was used), and the used numcepts are indicated before. "Reuters" correspond to "topic" and "local" to "intermediate local." For

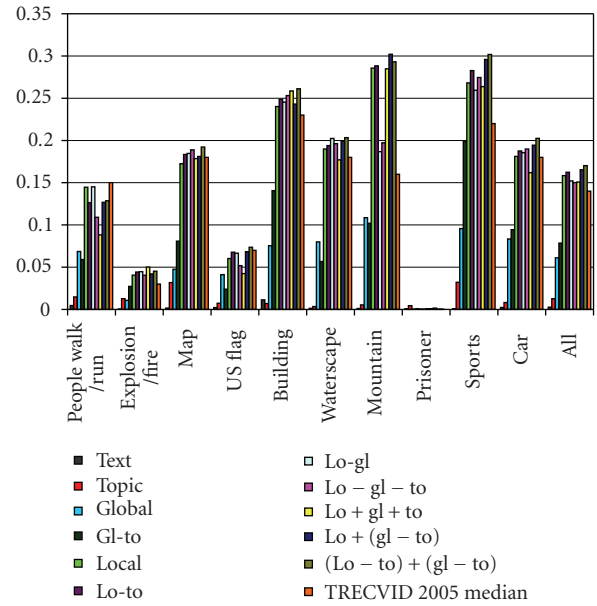

Figure 4: Mean average precision of the 10 high-level concepts of TRECVID 2005.

Table 2: Inferred average precision for the high-level feature extraction task; the dash means not within the official evaluation but evaluated in the same conditions.

Number	Run	IAP
1	Local-reuters-scale	0.0884
2	Local-text-scale	0.0864
3	Local-reuters-kernel-sum	0.0805
4	Local-reuters-kernel-prod	0.0313
5	Optimized-fusion-all	0.0674
6	Local-reuters-late-context	0.0753
—	local-reuters-early	0.0735
—	local-reuters-late	0.0597
—	local-text-early	0.0806
—	local-text-late	0.0584
—	local	0.0634
—	reuters	0.0080
—	text	0.0106

kernel fusion, two combining functions were used and indicated after the fusion scheme. The "scale" fusion scheme is an early fusion scheme in which the normalization before the SVM tool is done in such a way that each modality is given a weight that compensate for the unbalanced number of components in the input vectors. Finally, an "optimized" run which is a selection by feature of the scheme that best performed for that feature in the training set. It corresponds to a network in which a final layer has been added in which the last operator is a multiplexer controlled by the performance result on the training set.

6.3.1. *Unimodal runs*

We observe that the visual and text-based unimodal runs are very different in terms of accuracy; the visual-based classification is about 6 times better than the best text-based concept detection. This is probably due to the nature of the assessed concepts which seems to be hard to detect using text modality. This point is actually interesting for the evaluation of the ability of the various fusion schemes to handle such heterogeneous data. The features we want to merge lead to different accuracies and are also imbalanced regarding the number of input features.

6.3.2. *Classic early and late fusion schemes*

The two classical fusion schemes do not merge unimodal features similarly. While early fusion is able to outperform both unimodal runs, the late fusion scheme achieves poorer accuracy than the visual run. It might be due to the low number of dimensions handled by the stacked classifier. The early fusion scheme exploits context provided by all of the local visual features and the textual features. The gain obtained by such fusion means that those two modalities provide distinct kind of information. The merged features are, somehow, complementary.

6.3.3. *Early based fusion schemes*

The gain obtained by the normalized fusion schemes is the most important compare to other fusion schemes. Processing the unimodal features by reequilibrating them according to the number of dimensions is determinant in order to significantly outperform unimodal runs. In such a way, despite the different number of dimensions, both the visual and textual modalities have the same impact on concept classification.

This normalization process leads to a gain of almost 17% (in IAP) comparing to the classic early fusion scheme, which simply normalizes input in a common range, and 28% comparing to the better unimodal run.

The gain obtained by the kernel fusion scheme is less significant than the gain obtained by the normalized fusion run. However, when comparing to the classic early fusion, it seems that a combination using sum operator leads to better accuracy than multiplying kernels (which is somehow what the classic early fusion do). Furthermore, it is important to notice that the σ parameters are selected first by cross-validation on unimodal kernels and that we optimize then separately the linear combination. We can expect that an integrated framework which learns simultaneously σ_m (σ of modality m) and w_m (weight of modality m) parameters should lead to better results.

6.3.4. Contextual-late fusion scheme

Contextual-Late fusion is directly comparable with the classical late fusion scheme. This fusion scheme take into account the context from the score of other concepts detected in the same shot. By doing so, the context from other concepts leads to a gain of 26%. Furthermore, we observe that the MIAP obtained using the late contextual fusion scheme is almost the same as the one obtained for the classical early fusion scheme. In order to go further in this study, it could be interesting to evaluate the impact of the number and/or accuracy rate of concepts used in the context.

We notice that both of unimodal runs lead to poorer accuracy than the median of TRECVID 2006 participants. This may be due to the basic and not so optimized features used in our experiments. However, the gain induced by the three fusion schemes presented in this paper lead to better accuracy than the median. We think that an optimization in the choice of descriptors for each modality could enhance the accuracy rate of both unimodal and multimodal runs.

7. CONCLUSION

We have presented a framework for the design of concept detection systems for image and video indexing. This framework integrates in a homogeneous way all the data and processing types. The semantic gap is crossed in a number of steps, each producing a small increase in the abstraction level of the handled data. All the data inside the semantic gap and on both sides included are seen as a homogeneous type called *numcept* (covering from numbers to concepts). Similarly, all the processing modules between the various numcepts are seen as a homogeneous type called *operator*. Concepts are extracted from the raw signal using networks of operators operating on numcepts. These networks can be represented as data-flow graphs and the introduced homogenizations allow fusing elements regardless of their nature. Low-level descriptors can be fused with intermediate of final concepts.

This framework has been used to build a variety of indexing networks for images and videos and to evaluate many aspects of them. Using annotated corpora and protocols of the 2003 to 2006 TRECVID evaluation campaigns, we measured the benefit brought by the use of individual features, the use of several modalities, the use of various fusion strategies, and the use of topologic and conceptual contexts. The framework proved its efficiency for the design and evaluation of a series of network architectures while factorizing the training effort for common subnetworks.

As it is observed in the context of the TRECVID evaluation campaigns, the trend is to use such types of networks, to integrate as many features as possible, to use training sets as large and as rich as possible, and to design more and more sophisticated networks. Progress will continue to come with an increase in complexity, and this will be a challenge because of the combinatorial explosion of the possibilities of choosing and combining numcepts and operators. Learning operator networks via automatic generation and evaluation could be a good way of solving it. There will be a need for powerful tools for describing, handling, executing, and evaluating all these possible architectures. One possibility for that is to use the formalism of *functional programming* [11] over numcepts and operators. This formalism already proved to be efficient for the description of graphs of operators in the field of image processing [12].

ACKNOWLEDGMENTS

This work has been supported by the ISERE CNRS ASIA-STIC project and the Video Indexing INPG BQR project.

REFERENCES

[1] G. Iyengar, H. J. Nock, C. Neti, and M. Franz, "Semantic indexing of multimediq using audio, text and visual cues," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '02)*, Lausanne, Switzerland, August 2002.

[2] G. Iyengar and H. J. Nock, "Discriminative model fusion for semantic concept detection and annotation in video," in *Proceedings of the 11th ACM International Conference on Multimedia (MULTIMEDIA '03)*, pp. 255–258, Berkeley, Calif, USA, November 2003.

[3] A. Hauptman, R. V. Baron, M.-Y Chen, et al., "Informedia at TRECVID 2003 : analyzing and searching broadcast news video," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '03)*, p. 15, Gaithersburg, Md, USA, November 2003.

[4] M. R. Naphade and J. R. Smith, "On the detection of semantic concepts at TRECVID," in *Proceedings of the 12th ACM International Conference on Multimedia (MULTIMEDIA '04)*, pp. 660–667, New York, NY, USA, 2004.

[5] M. R. Naphade, "On supervision and statistical learning for semantic multimedia analysis," *Journal of Visual Communication and Image Representation*, vol. 15, no. 3, pp. 348–369, 2004.

[6] T.-S. Chua, S.-Y. Neo, Y. Zheng, et al., "TRECVID 2006 by NUS-I2R," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '06)*, Gaithersburg, Md, USA, November 2006.

[7] S. Ayache, G. Quénot, and S. Satoh, "Context-based conceptual image indexing," in *Processing of the IEEE International Conference on Acoustics, Speech and Signal Proceedings (ICASSP '06)*, vol. 2, pp. 421–424, Toulouse, France, May 2006.

[8] C. G. M. Snoek, M. Worring, and A. G. Hauptmann, "Learning rich semantics from news video archives by style analysis,"

ACM Transactions on Multimedia Computing, Communications and Applications, vol. 2, no. 2, pp. 91–108, 2006.

[9] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders, "The semantic pathfinder: using an authoring metaphor for generic multimedia indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1678–1689, 2006.

[10] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[11] J. Backus, "Can programming be liberated from the von Neumann style? A functional style and its algebra of programs," *Communications of the ACM*, vol. 21, no. 8, pp. 613–641, 1978.

[12] B. Zavidovique, J. Sérot, and G. M. Quénot, "Massively parallel dataflow computer dedicated to real time image processing," *Integrated Computer-Aided Engineering*, vol. 4, no. 1, pp. 9–29, 1997.

[13] S. Kumar and M. Hebert, "Discriminative random fields: a discriminative framework for contextual interaction in classification," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1150–1157, Nice, France, October 2003.

[14] M. R. Naphade, T. Kristjansson, B. Frey, and T. S. Huang, "Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia systems," in *Proceedings of International Conference on Image Processing (ICIP '98)*, vol. 3, pp. 536–540, Chicago, Ill, USA, October 1998.

[15] S. Ayache, G. Quénot, J. Gensel, and S. Satoh, "Using topic concepts for semantic video shots classification," in *Proceedings of 5th International Conference on Image and Video Retrieval (CIVR '06)*, vol. 4071 of *Lecture Notes in Computer Science*, pp. 300–309, Tempe, Ariz, USA, July 2006.

[16] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA '05)*, pp. 399–402, Singapore, November 2005.

[17] S. Ayache, G. Quénot, and J. Gensel, "CLIPS-LSR experiments at TRECVID 2006," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '06)*, Gaithersburg, Md, USA, November 2006.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] P. Over, T. Ianeva, W. Kraaij, and A. F. Smeaton, "TRECVID 2005—an overview," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '05)*, Gaithersburg, Md, USA, November 2005.

[20] M. Naphade, J. R. Smith, J. Tesic, et al., "Large-scale concept ontology for multimedia," *IEEE Multimedia*, vol. 13, no. 3, pp. 86–91, 2006.

[21] S. Ayache, G. Quénot, and J. Gensel, "Classifier fusion for SVM-based multimedia semantic indexing," in *Proceedings of 29th European Conference on Information Retrieval Research (ECIR '07)*, vol. 4425 of *Lecture Notes in Computer Science*, Rome, Italy, April 2007.

[22] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, and F. J. Seinstra, "The mediamill TRECVID 2004 semantic video search engine," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '04)*, Gaithersburg, Md, USA, November 2004.

[23] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[24] G. M. Quénot, "Computation of optical flow using dynamic programming," in *IAPR Workshop on Machine Vision Applications*, pp. 249–252, Tokyo, Japan, November 1996.

[25] C.-Y. Lin, B. L. Tseng, and J. R. Smith, "Video collaborative annotation forum: establishing groundtruth labels on large multimedia datasets," in *Proceedings of the TREC Video Retrieval Evaluation (TRECVID '03)*, Gaithersburg, Md, USA, November 2003.

[26] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: a new benchmark collection for text categorization research," *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[27] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," in *Proceedings of the Pacific Symposium on Biocomputing (PSB '04)*, pp. 300–311, Big Island of Hawaii, Hawaii, USA, January 2004.

[28] P. H. Gosselin and M. Cord, "A comparison of active classification methods for content-based image retrieval," in *Proceedings of the 1st International Workshop on Computer Vision Meets Databases (CVDB '04)*, pp. 51–58, Paris, France, June 2004.